

Received 16 June 2019; revised 15 July 2019; accepted 17 July 2019. Date of publication 30 July 2019; date of current version 30 September 2019.
The review of this paper was arranged by Editor M. Liu.

Digital Object Identifier 10.1109/JEDS.2019.2931757

Design and Test of the In-Array Build-In Self-Test Scheme for the Embedded RRAM Array

XIAOLE CUI¹, MIAOMIAO ZHANG¹, QIUJUN LIN¹, XIAOXIN CUI² (Member, IEEE), AND ANQI PANG¹

¹ Key Laboratory of Integrated Microsystems, Peking University Shenzhen Graduate School, Shenzhen 518055, China

² Institute of Microelectronics, Peking University, Beijing 100871, China

CORRESPONDING AUTHOR: X. CUI (e-mail: cuixx@pku.edu.cn)

This work was supported in part by the Shenzhen Science and Technology Innovation Committee under Grant JCYJ20170412150411676, and in part by the National Natural Science Foundation of China under Grant U1613215.

ABSTRACT An in-array build-in self-test (BIST) scheme is proposed for the embedded resistive random access memory (RRAM) array. The BIST circuit consists of the linear-feedback-shift-register (LFSR)-based pattern generator and the multi-input signature register (MISR)-based response compactor, and both the n -stage LFSR and MISR are implemented by $n + 2$ in-array RRAM cells. The proposed LFSR/MISR circuit has better performance than the IMPLY-based counterpart, due to the application of the proposed three-cycle XOR gate and two-cycle shift gate with the in-array RRAM cells. And it is more area efficient comparing with the memristor ratioed logic (MRL)-based counterpart. The proposed n -stage LFSR/MISR circuit is tested by the scan chain method. The test method only has the linear time complexity. For the best of our knowledge, it is the first attempt to design the in-array BIST circuit for the RRAM array.

INDEX TERMS RRAM test, LFSR, MISR, in-array BIST.

I. INTRODUCTION

The RRAM (Resistive Random Access Memory) device, which represents logic states with different resistance states, is an emerging non-volatile memory device compatible with the CMOS technology [1]. The test of the embedded RRAM array demands for the BIST (Build-In Self-Test) scheme due to the limited test accessibility. However, the BIST circuit, which is usually deployed outside the memory array, introduces extra area and pins.

In recent years, the researchers have found out that the RRAM device is a useful logic device [2]–[9], in addition to being used as memory device. Several RRAM based logic gate families, e.g., IMPLY (material implication) [2], CRS (complementary resistive switches) [3], MAGIC (memristor aided logic) [4], MRL (memristor ratioed logic) [5], LTG (linear threshold gate) [6], MAD (memristor as drivers) [7], Scouting [8], four-step logic [9], have been proposed. It inspires us to design the BIST circuit in the RRAM array itself, hoping to reduce the area overhead with the reasonable cost of performance. In the summary of BIST architectures, von de Goor [10] has pointed out that the LFSR (Linear Feedback Shift Register) is the most area efficient pattern generator module,

and the MISR (Multiple Input Signature Register) based response compactor can be applied to any combinations of the sub-arrays and the number of bits per access in the BIST. This work proposes an implementation of in-array BIST architecture with the LFSR based pattern generator and the MISR based response compactor, as presented as Fig. 1.

II. THE XOR AND SHIFT GATES

The LFSR, which consists of D flip flops and some XOR gates, generates the pseudorandom pattern sequence if it is initialized by a nonzero seeding pattern. The n -stage LFSR has n D flip flops, as shown in Fig. 1, the corresponding branch and XOR gate are deleted, if the Boolean parameter $h_i = 0$, $i \in [1, n - 1]$. The specific LFSR is defined by its characteristic polynomial. If the LFSR is described by a primitive polynomial, it generates the pseudorandom pattern sequence with the longest period, which contains $2^n - 1$ different patterns [11]. The state transition matrix T_s is presented in the formula (1). Assuming that the current states and the next states of the n -stage LFSR are $X_i(t)$ and $X_i(t + 1)$, respectively, $i \in [1, n - 1]$, the state transition is

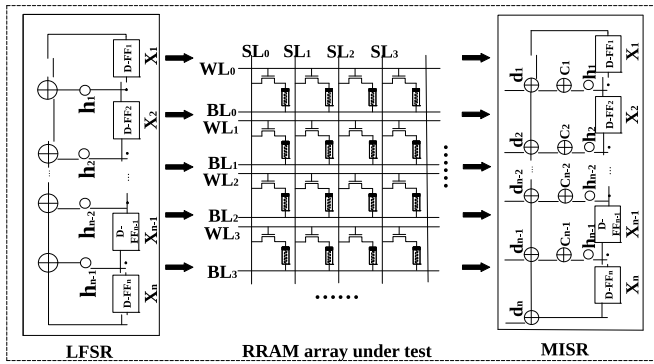


FIGURE 1. The overall BIST architecture.

described by the formula (2).

$$T_s = \begin{bmatrix} 0 & 1 & 0 & \dots & 0 & 0 \\ 0 & 0 & 1 & \dots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \dots & 1 & 0 \\ 0 & 0 & 0 & \dots & 0 & 1 \\ 1 & h_1 & h_2 & \dots & h_{n-2} & h_{n-1} \end{bmatrix} \quad (1)$$

$$X_i(t+1) = T_s \cdot X_i(t) \quad (2)$$

The n-stage MISR has n inputs and n D flip flops. The formula (3) describes the state transition of the n-stage MISR [11], where $d_i(t)$ is the current i^{th} input bit. It compacts the output sequences into a signature, i.e., the vector in the D flip flops. The MISR is a good test response compactor, for the aliasing probability of the signatures is only $1/(2^n)$ [11].

$$X_i(t+1) = T_s \cdot X_i(t) + d_i(t) \quad (3)$$

The RRAM based XOR and shift gates are required to build the in-array LFSR/MISR circuit. In the previous RRAM based logic gate families, only few of them are array implementable. The IMPLY logic is the first published logic gate [2]. It is logic complete but time consuming. The IMPLY based XOR gate consumes 8 cycles with 7 RRAM cells. All the CRS logic gates can be implemented in the RRAM array, but the gates need the special complementary RRAM cell [3]. For the MAGIC gate family, only the NOR and NOT gates are implementable in the RRAM array, and the XOR gate consumes 6 cycles with 7 RRAM cells [4].

The LFSR and MISR are the rich XOR gate circuit modules. We propose the improved XOR gate and shift gate for better circuit performance and less number of RRAM cells. The working steps of these two proposed gates are presented in Table 1. Each gate is realized by multiple steps, and each step consumes single cycle. The input logic states of the gates are presented by the input voltages at the input step, and the output logic states of the gates are presented by the resistance states of the output cell at the computation step. The high resistance state (HRS) represents the logic 0, and the low resistance state (LRS) represents the logic 1.

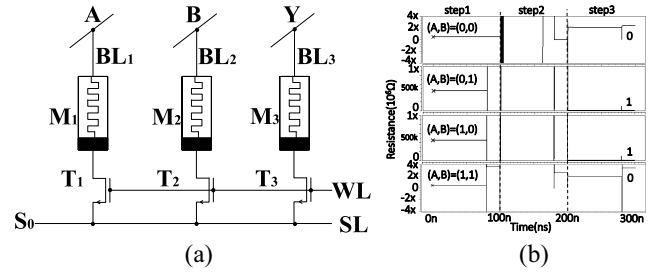


FIGURE 2. The proposed XOR gate (a) The XOR circuit (b) The simulation result on resistance value of M_3 .

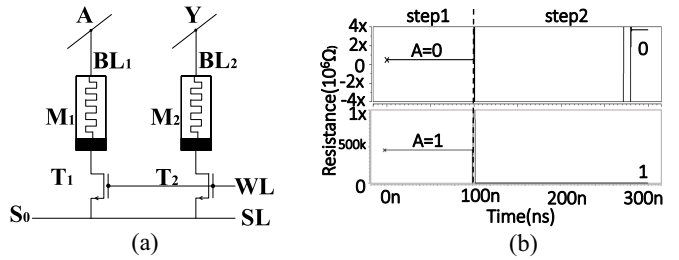


FIGURE 3. The proposed shift gate (a) The shift circuit (b) The simulation result on resistance value of M_2 .

The proposed XOR gate is shown in Fig. 2(a), where M_1 and M_2 are the input cells, M_3 is the output cell, and $T_1 \sim T_3$ are the selectors of the corresponding cells. The set end of the RRAM cell is labeled with the black thick line. The XOR gate works as follows.

Step 1 (Initialization): The voltage $V_{\text{init}} > |V_{\text{reset}}|$ is applied to the terminals A, B, and Y, where V_{reset} is the threshold voltage to turn the RRAM cell to HRS. The node S_0 is grounded. It resets $M_1 \sim M_3$ to HRS.

Step 2 (Input): Apply the input voltage V_{input} to the terminals A and B, respectively. $V_{\text{input}} = 0V$ for logic 0, whereas $V_{\text{input}} = V_{\text{high}} > 2V_{\text{set}}$ for logic 1, where V_{set} is the threshold voltage to turn the RRAM cell to LRS. The gate inputs configure the resistance state of the input cells, as described in Table 1.

Step 3 (Computation): Connect the output terminal Y to GND, and apply the computation voltage V_c to the terminals A and B, where $V_{\text{set}} < V_c < \min[1.5V_{\text{set}}, 2|V_{\text{reset}}|]$, then the XOR result is presented as the resistance state of M_3 . The resistance states of $M_1 \sim M_2$ remain stable during this step even if M_3 is changed to LRS, because $V_c < \min[1.5V_{\text{set}}, 2|V_{\text{reset}}|]$.

The proposed shift gate is shown in Fig. 3(a), where M_1 is the input cell and M_2 is the output cell. This circuit copies the logic state of the gate input to the output cell. The shift gate works as follows.

Step 1 (Initialization): Reset M_1 and M_2 to HRS.

Step 2 (Computation): Apply the input voltage to the terminal A, $V_{\text{high}} > 2V_{\text{set}}$ for logic 1, and GND for logic 0. The output terminal Y is connected to GND. If the input state is logic 1, M_2 is set to LRS; M_2 remains HRS if the input state is logic 0.

TABLE 1. The working steps of the proposed XOR and shift gates.

Gate	Steps	A	B	S0	Y	State of the Output Cell	Functionality
XOR	Initialization	V_{init}	V_{init}	GND	V_{init}	HRS	Reset $M_1 \sim M_3$ to HRS. $M_1 \sim M_2$ keep the HRS if $(A,B)=(0,0)$ or $(1,1)$; M_1 and M_2 are set to the LRS if $(A,B)=(0,1)$ and $(A,B)=(1,0)$, respectively.
	Input	V_{input}	V_{input}	Floating	Floating	HRS	
	Computation	V_c	V_c	Floating	GND	Output State	Output the result by the resistance state of M_3 .
Shift	Initialization	V_{init}	--	GND	V_{init}	HRS	Reset $M_1 \sim M_2$ to HRS.
	Computation	V_{input}	--	Floating	GND	Output State	Output the result by to the resistance state of M_2 .

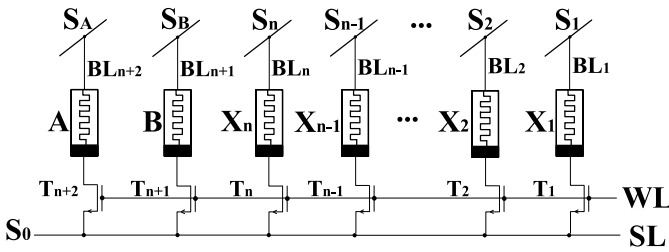


FIGURE 4. The n-stage standard LFSR / MISR circuit.

The proposed gate circuits are simulated with the RRAM device model [12] and the SPICE tool. The simulation conditions are $V_{set} = 0.8V$, $V_{reset} = -1.2V$, $V_{init} = 1.7V$, $V_{high} = 1.7V$, $V_c = 1V$. Fig. 2(b) and Fig. 3(b) show the simulation results on the resistance value of the output cell for the XOR gate and shift gate, respectively, where is number of working steps are labeled on the top of the Figures. The maximum resistance value of logic 1 in the last step is about $5.44 \times 10^3 \Omega$, whereas it is at least $2.37 \times 10^6 \Omega$ for the logic 0. Although the resistance of the output cell varies in each cycle, the final results meet the functional requirements of the proposed XOR gate and shift gate.

III. THE BIST CIRCUIT

Fig. 1 presents a common BIST architecture. We design the in-array BIST circuit by realizing the LFSR and MISR modules with the XOR and shift gates in the RRAM array itself.

The proposed general n-stage standard LFSR circuit is presented in Fig. 4. It consists of n+2 RRAM cells, where A and B are the input cells of the XOR gates, and $X_1 \sim X_n$ store the logic states. Assumes that the seeding data are $D_{s1} \sim D_{sn}$, and the corresponding voltages are $V_1 \sim V_n$. The n-stage LFSR circuit with multiple XOR gates works following the steps in Table 2.

Phase 1: Initialization. Reset all the RRAM cells into HRS.

Phase 2: Initialize the RRAM cell $X_{n-1} \sim X_1$ by the shift operations with the seeding data $D_{s(n-1)} \sim D_{s1}$, respectively. The detailed operations are described in Table 2. It consumes n-1 cycles.

Phase 3: Realize the XOR computations. It uses the X_n as the output cell for the XOR operations, and the XOR functions in the LFSR are executed sequentially. For each XOR operation, it fetches the input data from the corresponding cells, then applies them to the XOR gate to obtain the result in X_n . To prepare for the next XOR operation, the state of X_n is readout, and the cell A, B, and X_n are reset to HRS. It is seen from Table 2 that the first XOR gate consumes 4 cycles, and every other XOR gate consumes 5 cycles. The number of cycles to obtain the first LFSR output is determined by the number of XOR gates in the feedback path. For the slowest case, in which there are n-1 XOR gates in the feedback path, it needs $6n - 6$ cycles. Whereas for the fastest cases, in which there is only one XOR gate in the feedback path, it only consumes $n + 4$ cycles to generate the first pattern.

Phase 4: Readout the current pattern as the test inputs of the RRAM array.

Phase 5: Shift operations of LFSR. It firstly resets the cell A, B, and X_1 , then executes the read, reset, and shift operations. It is seen from Table 2 that the X_1 is updated with 2 cycles, and the $X_i, i \in [2, n - 1]$ is updated with 3 cycles. The n-stage LFSR consumes $3n-3$ cycles to complete this phase.

Then the XOR, read, and shift phases work alternatively till the end. Each pattern are obtained after one shift phase and one XOR phase, it consumes $8n - 8$ cycles for the slowest case, and $3n + 2$ cycles for the fastest cases.

Table 3 compares the area and performance of several RRAM based LFSR circuits. The circuit area is presented by the number of RRAM cells and MOS transistors, and the circuit performance is characterized by the number of cycles to generate one new pattern. Obviously, the proposed LFSR circuit has better performance comparing with the IMPLY based counterpart [13]. And the proposed in array LFSR circuit is more area efficient comparing with the MRL based LFSR scheme [14], though it consumes more working cycles.

The BIST designers usually implement two n-stage LFSR circuits with reciprocal characteristic polynomials as the pattern generator to generate the up and down pattern sequences. Both the two LFSR circuits can share the $n + 2$ RRAM cells. The summation of cycles needed for one pattern in the up and down pattern sequences is constantly $8n - 8$, because

TABLE 2. The working steps of the proposed LFSR circuit with multiple XOR gates.

Phase	Steps	Operations	Related RRAM cells	Description
Initialization	1	V_{init} is applied to S_A, S_B , and $S_1 \sim S_n$, S_0 is grounded	All the cells in Fig.4	Reset A, B, and $X_1 \sim X_n$ to HRS
	2	V_{n-1} is applied to terminal S_A , S_{n-1} is grounded	A and X_{n-1}	$D_{s(n-1)} \rightarrow X_{n-1}$
Seeding	3	V_{n-2} is applied to terminal S_A , S_{n-2} is grounded	A and X_{n-2}	$D_{s(n-2)} \rightarrow X_{n-2}$

	n	V_1 is applied to terminal S_A , S_1 is grounded	A and X_1	$D_{s1} \rightarrow X_1$
XOR	n+1	Read X_1 , convert the logic state to V_1	X_1	$X_1 \rightarrow V_1$
	n+2	Read X_2 , convert the logic state to V_2	X_2	$X_2 \rightarrow V_2$
	n+3	Apply V_1 and V_2 to S_A and S_B respectively	A, B	Input step of XOR gate
	n+4	V_c is applied to S_A and S_B , S_n is grounded	A, B, X_n	$X_1 \text{ XOR } X_2 \rightarrow X_n$
	n+5	Read X_n , convert the logic state to V_x	X_n	$X_n \rightarrow V_x$
	n+6	V_{init} is applied to S_A, S_B , and S_n	A, B, X_n	Reset A, B, and X_n to HRS.
	n+7	Read X_3 , convert the logic state to V_3	X_3	$X_3 \rightarrow V_3$
	n+8	Apply V_3 and V_x to S_A and S_B respectively	A, B	Input step of XOR gate
	n+9	V_c is applied to S_A and S_B , S_n is grounded	A, B, X_n	$X_1 \text{ XOR } X_2 \text{ XOR } X_3 \rightarrow X_n$

	6n-8	Read X_{n-1} , convert the logic state to V_{n-1}	X_{n-1}	$X_{n-1} \rightarrow V_{n-1}$
	6n-7	Apply V_{n-1} and V_x to S_A and S_B respectively	A, B	Input step of XOR gate
6n-6	V_c is applied to S_A and S_B , S_n is grounded	A, B, X_n	$X_1 \text{ XOR } X_2 \text{ XOR } \dots \text{ XOR } X_{n-1} \rightarrow X_n$	
Read	6n-5	Readout the current pattern	$X_1 \sim X_n$	Read $X_1 \sim X_n$ as the test inputs
	6n-4	V_{init} is applied to S_A, S_B, X_1	A, B	Reset A, B, X_1
Shift	6n-3	Read X_2 , convert the logic state to V_x	X_2	$X_2 \rightarrow V_x$
	6n-2	V_x is applied to terminal S_A , S_1 is grounded	A, X_1	$X_2 \rightarrow X_1$
	6n-1	Read X_3 , convert the logic state to V_x	X_3	$X_3 \rightarrow V_x$
	6n	V_{init} is applied to X_2	X_2	Reset X_2 to HRS
	6n+1	V_x is applied to terminal S_A , S_2 is grounded	A, X_2	$X_3 \rightarrow X_2$

	9n-9	V_{init} is applied to X_{n-1}	X_{n-1}	Reset X_{n-1} to HRS
	9n-8	V_x is applied to terminal S_A , S_{n-1} is grounded	A, X_{n-1}	$X_n \rightarrow X_{n-1}$
XOR	9n-7	Read X_1 , convert the logic state to V_1	X_1	$X_1 \rightarrow V_1$

of the reciprocal feature. The all-zero pattern is generated in the initialization phase in Table 2, it only needs one additional read operation after the reset operation in the array test.

The MISR is able to obtain a very low aliasing rate if the number of stage is great enough. Actually, the proposed circuit in Fig. 4 can also implement the n-stage MISR function with extra 5n cycles because of the additional n XOR gates. The slowest and the fastest n -stage MISR consume $13n - 8$ and $8n + 2$ cycles, respectively, to generate one new signature.

The power of the reset, set and read operations are $15.9375 \mu\text{W}$, $21.4806 \mu\text{W}$, and $0.3815 \mu\text{W}$, respectively, with the simulation conditions above. And the average power of XOR operation and shift operation at 10MHz are $36.1684 \mu\text{W}$ and $13.8415 \mu\text{W}$, respectively, which include the power of the read operations. The two reciprocal LFSR consumes $n - 1$ XOR operations, $2(n - 1)$ shift operations, the total power of the pattern generator is $63.8514n - 63.8514 \mu\text{W}$ @ 10MHz. Similarly, the maximum n-stage MISR consumes $86.1783n - 50.0099 \mu\text{W}$ @10MHz. The proposed BIST circuit totally consumes

TABLE 3. Comparisons on the area and performance of the N-stage RRAM based LFSR circuits.

LFSR circuits	IMPLY[13]	MRL [14]	Proposed
RRAM cells	n+4	16n+4	n+2
MOS transistors	0	16n+3	n+2
number of cycles for the	11(n+1)	2n+1	3n+2
LFSR with one XOR gate			
number of cycles for the	11(2n-1)	3n-1	8n-8
LFSR with n-1 XOR gates			

150.0297n – 113.8613 μW @10MHz averagely to generate one pattern in the up and down pattern sequences.

The XOR gate and D flip flop consume 52.349μW and 12.945 μW @10MHz, respectively, with the 65nm CMOS technology file. The BIST circuit with two reciprocal n-stage LFSR and one n-stage MISR consumes 195.882n – 143.533 @10MHz in total to generate one pattern in the up and down pattern sequences. Whereas, the XOR gate and D flip flop consume 1.4297μW and 1.7244 μW, respectively, with the 28nm technology file at 10MHz. The BIST circuit with two reciprocal n-stage LFSR and one n-stage MISR consumes 9.4623n – 8.0326 μW @10MHz in total with the 28nm CMOS technology. The power consumption of the proposed in-array BIST circuit is roughly in the same order of the power of the BIST circuit with the 65nm CMOS technology. And the proposed BIST circuit consumes more power than the counterpart with more advanced CMOS technology.

IV. TEST OF THE BIST CIRCUIT

The RRAM cells in the array under test are divided into two groups. The group A contains the RRAM cells for the LFSR and MISR modules, whereas the other RRAM cells in the array are in the group B. The proposed BIST circuit in the array tests the RRAM cells in group B, only if the RRAM cells in group A are proved fault free by tests. Both the n-stage LFSR and MISR consume n + 2 RRAM cells. These RRAM cells in group A can be tested by configuring them into a scan chain. By inputting the test sequence 00110 into the scan chain, the function of n + 2 RRAM cells is tested by observing the results of the shift operations at the tail cell of the scan chain. One additional cell X_{in} is required for the input cell of the shift gate. The shift gate works if X_{in} can be reset to HRS successfully, so the X_{in} should be stuck-at 1 fault free. It is tested by one write 0 operation and one read 0 operation. If X_{in} is stuck-at 1 fault free, the n+2 RRAM cells, n + 2 ≥ 5, for the LFSR or MISR circuit are tested by the three phases shown in Fig. 5, where X₁ and B are the heading cell and tail cell of the scan chain, and X_{in} is the input cell of the shift gate.

Phase 1: Input the test pattern I1 ~ I5 = 00110. It firstly resets all the n + 3 RRAM cells into HRS, then scans in the test pattern to X1 ~ X5 with shift operations. As shown in Fig. 5 (a), the 00110 is fed into the scan chain by

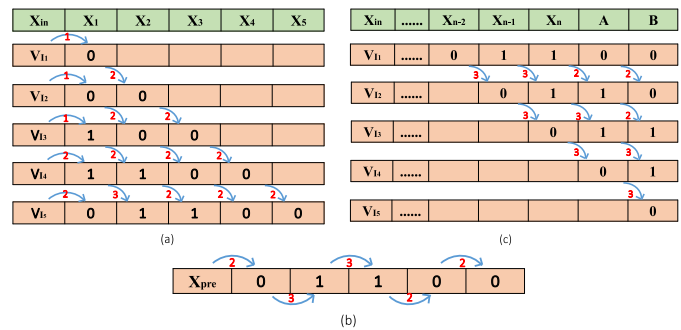


FIGURE 5. The scan test of LFSR/MISR circuit.

(1 + 2 + 3 + 4 + 5) = 15 shift operations. It is seen from Table 2 that, the value of the source cell is shifted into the scan chain by (1 + 2 + 3 + 4 + 5) = 15 shift operations. It is seen from Table 2 that, the value of the source cell is shifted to the target cell after the read operation on the source cell and the reset operation on the target cell. Actually, for the shift operation for the heading cell X₁, the cycle for the read operation can be saved. Furthermore, if the previous value of the target cell is 0, i.e., the target cell is in HRS, the cycle for the reset operation can also be saved. So these shift operations only consume 27 cycles, and this phase consumes 28 cycles in total.

Phase 2: Shift I1 ~ I5 in the scan chain. The pattern sequence I1 ~ I5 shifts (n+2–5) times before the first output is observed at the tail cell of the scan chain. Actually, each shift operation of the pattern sequence 00110 requires 3 two-cycle shift operations and 2 three-cycle shift operations, i.e., 12 cycles, as presented in Fig. 5 (b). So the shift operations in this phase need 12 (n + 2 – 5) = 12n – 36 cycles, and this phase consumes 12n – 35 cycles, including the read operation.

Phase 3: Scan out. The rest of the data in the scan chain are shifted out by (4 + 3 + 2 + 1) = 10 shift operations, as shown in Fig. 5 (c). It consumes 31 cycles including the read operations.

The total cycle counts for the test of the n-stage LFSR/MISR circuit include the 2 cycles for the test of X_{in}, and the cycles for the scan test of the n + 2 cells. It requires 12n + 24 cycles, if n + 2 ≥ 5. For the n + 2 = 4 case, the scan test consumes only 48 cycles without the operations in phase 2. All the transition faults and stuck-at faults in the n + 2 RRAM cells for the BIST circuit are tested. The test complexity of the proposed BIST scheme is comparable to that of March C* [15] with the cost of lower fault coverage. Fortunately, the fault coverage of the proposed BIST scheme is acceptable, because the transition fault and the stuck-at fault have the highest occurrence frequency in the RRAM fault set [15].

V. CONCLUSION

We propose an in-array BIST scheme based on the non-volatile RRAM devices, which only consumes n+2 RRAM cells for the n-stage LFSR pattern generator and MISR

response compactor. The proposed BIST has good performance and small area, and it can be tested with an acceptable linear time complexity. The most significant characteristic is that it reduces the circuit area dramatically, because the relatively area consuming MOSFET BIST circuit outside array is not required if the proposed in-array RRAM BIST scheme is applied.

REFERENCES

- [1] D. B. Strukov, G. S. Snider, D. R. Stewart, and R. S. Williams, "The missing memristor found," *Nature*, vol. 453, pp. 80–83, May 2008.
- [2] J. Borghetti, G. S. Snider, P. J. Kuekes, J. J. Yang, D. R. Stewart, and R. S. Williams, "'Memristive' switches enable 'stateful' logic operations via material implication," *Nature*, vol. 464, pp. 873–876, Apr. 2010.
- [3] R. Rosezin, E. Linn, C. Kugeler, R. Bruchhaus, and R. Waser, "Crossbar logic using bipolar and complementary resistive switches," *IEEE Electron Device Lett.*, vol. 32, no. 6, pp. 710–712, Jun. 2011.
- [4] S. Kvatinsky *et al.*, "MAGIC—Memristor-aided logic," *IEEE Trans. Circuits Syst. II, Exp. Briefs*, vol. 61, no. 11, pp. 895–899, Nov. 2014.
- [5] S. Kvatinsky, N. Wald, G. Satat, A. Kolodny, U. C. Weiser, and E. G. Friedman, "MRL—Memristor ratioed logic," in *Proc. IEEE Int. Workshop Cellular Nanoscale Netw. Appl.*, Turin, Italy, 2012, pp. 1–6.
- [6] L. Gao, F. Alibart, and D. B. Strukov, "Programmable CMOS/memristor threshold logic," *IEEE Trans. Nanotechnol.*, vol. 12, no. 2, pp. 115–119, Mar. 2013.
- [7] L. Guckert and E. E. Swartzlander, "MAD gates—Memristor logic design using driver circuitry," *IEEE Trans. Circuits Syst. II, Exp. Briefs*, vol. 64, no. 2, pp. 171–175, Feb. 2017.
- [8] L. Xie *et al.*, "Scouting logic: A novel memristor-based logic design for resistive computing," in *Proc. IEEE Comput. Soc. Annu. Symp. VLSI*, Bochum, Germany, 2017, pp. 176–181.
- [9] K. Zhang, X. Cui, and X. Cui, "A design of high performance full adder with memristors," in *Proc. IEEE Int. Conf. ASIC*, Guiyang, China, 2017, pp. 746–749.
- [10] A. J. van de Goor, *Testing Semiconductor Memories: Theory and Practice*, Chichester, U.K.: Wiley, 1991, pp. 350–356.
- [11] M. L. Bushnell and V. D. Agrawal, *Essentials of Electronic Testing for Digital, Memory and Mixed-Signal VLSI Circuits*, Boston, MA, USA: Kluwer, 2000, pp. 495–536.
- [12] H. Li, P. Huang, B. Gao, B. Chen, X. Liu, and J. Kang, "A SPICE model of resistive random access memory for large-scale memory array simulation," *IEEE Electron Device Lett.*, vol. 35, no. 2, pp. 211–213, Feb. 2014.
- [13] M. Teimoory, A. Amirsoleimani, A. Ahmadi, S. Alirezaee, S. Salimpour, and M. Ahmadi, "Memristor-based linear feedback shift register based on material implication logic," in *Proc. IEEE Eur. Conf. Circuit Theory Design*, Trondheim, Norway, Aug. 2015, pp. 1–4.
- [14] A. Sasi, A. Amirsoleimani, A. Ahmadi, and M. Ahmadi, "Hybrid memristor-CMOS based linear feedback shift register design," in *Proc. IEEE Int. Conf. Electron. Circuits Syst.*, Batumi, Georgia, Dec. 2017, pp. 62–65.
- [15] C.-Y. Chen *et al.*, "RRAM defect modeling and failure analysis based on March test and a novel squeeze-search scheme," *IEEE Trans. Comput.*, vol. 64, no. 1, pp. 180–190, Jan. 2015.